

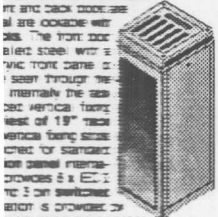
! KS
Savings always
wanted for cash!

RACK DINETS
super quality foot 40U
usually Ultra Smart
ss b. Half Price!
19" rack dinets made in UK by
Enclosures Ltd. Units feature
removable front door,
removable back door
removable side panels. Fully
internal fans, easy punched
configuration & equipment mounting
10" mounted racks. 2 way 13 amp
switched mains distribution strip make
this some of the most versatile we
be stacked side by side and therefore
save space in multiple bays.
H x 32" W x 22" D. Order as
removable dinets. £345.00 (G)
E dinets £245.00 (G)

shelves, accessories
wide 3 to 4" high.
from stock !!

ty - All steel RakCab

is up to the highest possible spec,
practices with removable



front and back doors are
all are removable
The front door
black steel with a
19" rack dinet
seen through the
mainly the rack
vertical facing
19" rack dinet
vertical facing
for sample
side panel inter-
connects & E.C.
to 3 on switches
error is provided
side of the section
the door may be removed for fitting
etc. Other features include: fitted
latched with a cover rear for
supported in excellent, slightly used
Rack Dinets. External dimensions
56" x 25" x 23" (H x W x D)

a third of others price !!

only £245.00 (G)

above only £345 - CALL

SCO 50% off !!

cancelled. Just order brings you

these ultra high spec 12v DC 14 Ah

by Hewlett Energy Ltd. type SBS15

offer a far superior shelf & guaran-

tee BT & BS6290 approved. Supplied

in 200 wide, 12" high, 77 deep.

Each current makes once over £70

each or 4 for £99 (G)

100 FROM STOCK

see also from our Massive Stocks

Cable, Cradle, hermetically Sealed,

Deep Red, Mercury Wetted, Solid

etc. CALL or see our web site

information. Many associate types from

CAMERAS

same technology &

quality product test-

COO camera at a

all automatic setting for

low light & high light

conditions. A 10 mm fixed focus

lens gives excellent focus

resolution from close up to long

distance. The composite video output will

be to any composite monitor or TV

SCART socket) and most video

cards. Unit runs from 12V DC so

for security & portable applica-

tions where mains power not available.

Size 117 deep x 43 high. Supplied

with user data, 100% of applica-

tion. Also TV, Web Cams etc. etc.

£99.00 or 2 for £180.00 (G)

SPECIALS

piece with service pack 3

ged. ONLY £89.00 (G)

latest - but at this price! £7.95

re books on QBasic £14.95

£55.00 on 3.5" disks

£24.95 on 5 1/4" disks with manual

or software is code B

v.distel.co.uk

ENQUIRIES

86 3333

AX 0208 653 8888

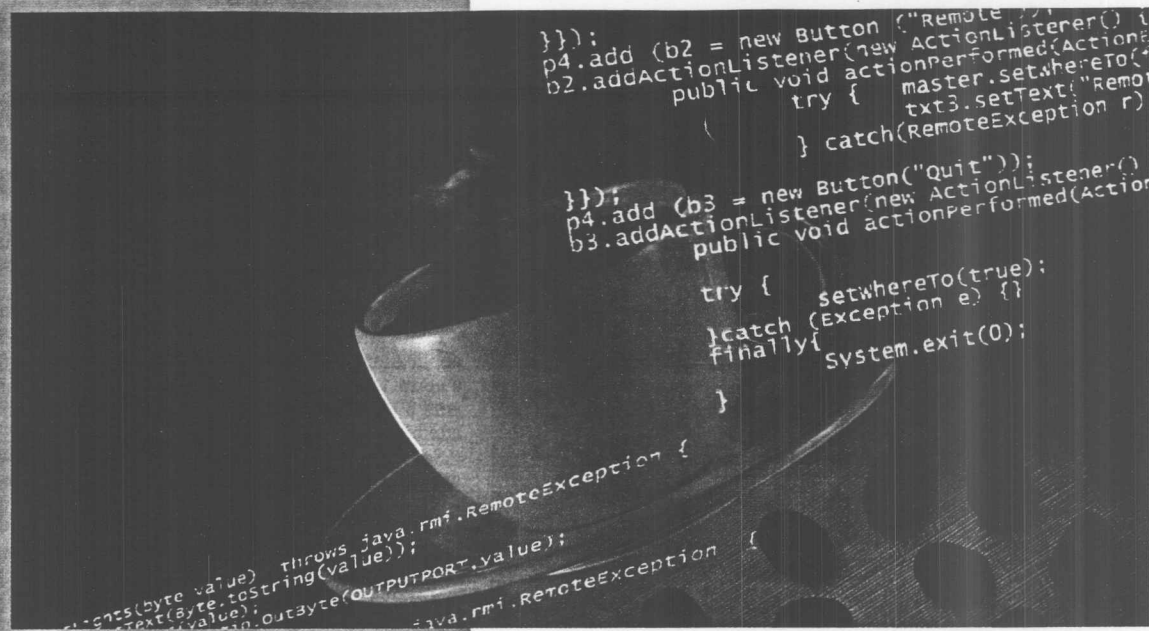
is supplied from Government, Schools,

armed forces (A1) £3.00, (A11) £4.00,

100% supplied to our Standard Conditions

terms without prior notice. Orders subject

to our terms & conditions.



Tini Java

Java expert Les Hughes has been experimenting with a tiny controller designed for Internet connection. Although very low cost, this controller is a complete computer with Internet, network and serial i/o capabilities, giving it huge potential for remote i/o and telemetry applications.

Much noise has been made concerning Java. Although originally conceived as a technology capable of powering embedded systems, it is only recently that Java devices have appeared on the market. One of the most interesting of these is the Tiny InterNet Interface – or TINI – from Dallas Semiconductors.

TINI executes a Java Virtual Machine, which in turn executes Java 'bytecodes' – in a similar manner to any other Java platform. However, although TINI runs Java, it is not a hardware implementation of a VM. Instead, the current TINI hardware is based on the Dallas 80C390 micro controller.

The TINI VM, operating system and various user programs are loaded into flash memory. This enables simple updates, which are freely downloadable from the project web site – www.ibutton.com/TINI.

The current TINI board incorporates a plethora of external interfaces: 10baseT ethernet, Dallas one-wire, I²C and RS232. Besides the processor, the TINI board also contains 512Kbyte of flash memory, 512Kbyte of NVRAM and an RTC. Processor address, data and control busses are brought out to the edge of the board for custom expansion – e.g. memory expansion or

custom i/o. That's quite a few features for a device that is no bigger than a normal PC SIMM and only costs \$50.00!

At the time of writing, TINI firmware was nearing beta 3. The processor board has been slightly redesigned to fit a more standard 72-pin SIMM socket rather than the original 68-pin connector. Although TINI is still beta for both hardware and firmware, recent releases have enhanced performance, stability and features.

Of course, as with any beta product, revisions often occur and APIs and specifications change, based on

Free software

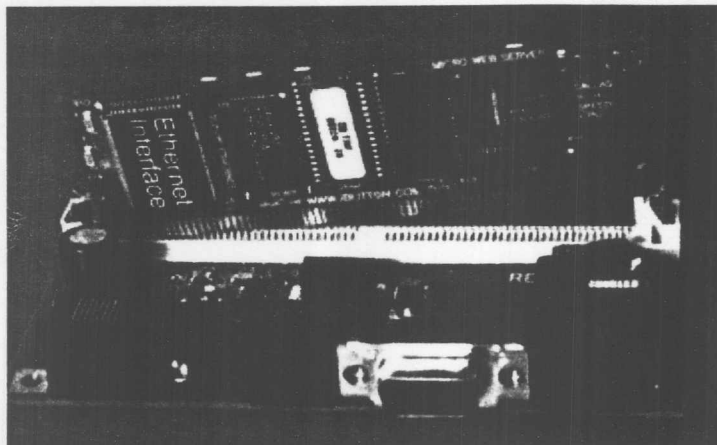
All of the software required to develop applications for TINI is available free of charge from various Internet sites. Installing and configuring your environment in order to get TINI up and running consists of a number of tasks. Guillaume Fournier's excellent guide at

<http://www3.sympatico.ca/guillaume.fournier/>

describes in detail the process that you should follow in order to be able to boot your TINI.

Les is a Technical Architect with Rubus.com – a fast growing e-Business Consultancy, currently working with a number of dot-com clients.

user feedback. In a similar manner to open-source software development, TINI has been subject to the input of a large community. This has enhanced the end product in a way that is not possible within a closed environment. Perhaps this open development is the most unusual aspect of the project.



TINI – for around \$50, you get 10baseT ethernet, Dallas one-wire, P^2C and RS232 interfaces, a processor, 512Kbyte of flash memory, 512Kbyte of NVRAM and an RTC. At its core is a soft Java virtual machine that's easy to update.

Establishing a network

While it is not actually necessary to network TINI in order to start experimenting, you'll be missing out on the whole idea that makes TINI so special – an embedded network node that runs Java.

Networking hardware has reduced in price dramatically in recent years. Browsing any of the monthly PC magazines shows a whole host of network cards for less than £15.00 and small 'micro' hubs – i.e. those with somewhere between 3 and 10 ports – for less than £30.00.

While it is beyond the scope of this article to delve into the intricacies of network engineering and the selection of networking hardware, we can examine the steps required to establish a simple TCP/IP network at home.

Of course you'll need a network card for your PC. I use a £10 NE2000 clone. Any 10baseT card should do, as long as it's supported by your chosen operating system.

You have two basic hardware options for connecting TINI to your personal computer. You can either use a crossover network cable or buy a small network hub and patch leads.* The crossover network cable has the advantage of being the cheapest but a small network hub offers far greater flexibility for not much more outlay.

Setting up your network

First, you'll need to ensure that you have TCP/IP networking installed for your chosen development platform – but this is the easy part as most systems come pre-configured with this option. If you dial into the Internet, you almost certainly have TCP/IP installed. If not, for a Windows platform, you can add this option in Control Panel -> Network -> Protocols.

Once you have all of the hardware installed and connected, you will need to configure each of the machines on the network with an address. If you use your PC for surfing the net then you should assign a network address from the range 192.168.0.0 to 192.168.255.255. These are from a range of special addresses that, if they 'escape' from your private network onto the Internet, can't cause any damage.

Addresses ending in '.0', '.1' and '.255' should not be used as they have special meaning on the network. For the rest of this article, I'll assume that you've chosen 192.168.0.50 for your PC and 192.168.0.100 for TINI.

Addresses are assigned to PCs in Control Panel -> Network. Find the TCP/IP protocol that maps to your network card – not to your dial-up network adapter! Select properties and then the IP address tab. Enter your chosen IP address in the box provided and 255.255.255.0 into the 'subnet

Getting started with TINI

In order to take full advantage of TINI's network abilities – it is the tiny Internet interface after all – you will need to connect TINI to a LAN. Many home users won't have their own IP network but this need not put you off. The box 'Establishing a network' shows how simple and cheap it is now to 'get wired'.

As has become traditional in the world of computing, our first TINI program is a network server version of the 'Hello World' classic. Groans aside, this simple application serves a number of purposes.

Firstly, the procedure to compile, build and load an application onto TINI is somewhat different from that of a normal Java application. A simple application can help with establishing that all is working correctly before we attempt something more extravagant.

Secondly, the program demonstrates a standard way of writing a multi-threaded network server. Next time, we'll be looking at something far more interesting but until then...

Listing 1 is the source code for our Hello Server. As mentioned, this program simply waits for a network connection, says Hello to the connecting machine, then closes the connection. Most Internet services such as web servers, etc., operate in a similar way.

Back to the example. Since this program uses Java's network and input/output libraries, these are imported at the top of the file. Next, define the name of the application – i.e. HelloServer. The main method forms the entry point to the program; this method is called by the

mask'. You can leave all of the other boxes empty. You can check your settings by using the commands winipcfg on windows 9x and ipconfig on NT.

TINI's IP address is set using the ipconfig command thus:

```
TINI /> ipconfig -a
192.168.0.100
```

You will need to login to your TINI board using JavaKit over a serial line in order to run ipconfig. Of course, you should have installed the firmware and booted TINI first!

Once you have two devices configured on the network, try pinging each one in turn from the other, e.g.,

```
C:\> ping 192.168.0.100
Pinging 192.168.0.100 with 32
bytes of data
Reply from 192.168.0.100:
bytes=32 time<10ms TTL=64
```

```
TINI /> ping 192.168.0.50
Got a reply from node
192.168.0.50/192.168.0.50
Sent 1 request(s), got 1
reply(s)
```

This shows that all's OK between the PC and TINI and vice versa.

*The author means a UTP crossover cable. Note that there are NE2000 clones that only have BNC connectors. Ed.

Java virtual machine when it starts our program.

Our main method defines a `Socket` field called `client`, which is used for incoming connections. However, in order to receive these connect requests from clients, we have to use a `'ServerSocket'` to manage the process.

A `ServerSocket` 'binds' to a particular 'port' and listens for connections. When a `ServerSocket` accepts a connect request from a distant client, it passes the connection on, in the form of a `Socket` object, and goes back to listening. In this way, you don't have to wait for a client to finish using the Server program before others can connect.

It's a bit like phoning your bank's call centre. You dial a single number and your call is routed through to any available operator, allowing more calls to come in.

In order to be able to process multiple connections simultaneously, you can take advantage of Java 'threads' and make our server multithreaded. This is what's happening in the statement new `HelloServer(client)`. Our main method takes the `Socket` returned by the `ServerSocket` and creates a new `HelloServer` object to handle the connection. This object automatically starts a new thread upon creation and starts talking to the client.

This action can be seen in the constructor methods,

```
serverThread = new
Thread(this);
serverThread.start();
```

method calls.

The `start()` method eventually calls `run()` method. The `run` method first asks the `Socket` for something to write to – the `OutputStream` – and then turns this into something that can be printed to. You then simply print a message to this `Writer`.

Once we've sent our message we wait for a second, to allow you to read the message and then, rather rudely, we close our output channel and the socket, thus cutting off the client before they can respond to our Hello.

Building the application

Once you have entered the `HelloServer` code, you will need to turn it into a format suitable for the TINI.

First, compile the `HelloServer.java` file using,

```
javac -bootclasspath
<TINIPATH>/tiniclasses.jar
HelloServer.java
```

replacing the `<TINIPATH>` tag with the location of your TINI installation, for example, `C:\tini`.

The `-bootclasspath` directive allows the compiler, which is written in Java, to use a different set of core classes (`java.lang`, `java.io`, `java.net`, etc..) from those compiled into the application. You won't be running under the standard JVM

So what can I do with it?

Interestingly, Tina's manufacturer, Dallas Semiconductor, is still a technology driven company. From the outsider's view – and this is often reflected in Dallas engineers' posts to the TINI mail list – it seems as if the company produces numerous clever solutions just waiting for a problem to come along.

TINI is more than just a rather cool toy though; it is a near-complete implementation of the J2ME platform albeit in pre-production, beta form.

At present, judging from the 'TINI-users' list, real-world TINI applications range from data loggers, security systems and network server monitors to GPS-aware systems and simple dial-up gateways. TINI could be applied in almost any scenario requiring a networked controller; from remote surveillance with an off-the-shelf webcam and wireless LAN technology to home automation using a DS1920 and Tesco Direct to ensure that you always have a good supply of cold beer!

Listing 1. `HelloServer.java`. This is a simple server that says 'Hello' to any clients.

```
/* HelloServer.java
*/
import java.io.*;
import java.net.*;

public class HelloServer implements Runnable {
    public static void main(String args[]) {
        Socket client;
        try{
            //Create a new server listening on port 1234
            ServerSocket server = new ServerSocket(1234);
            while(true){
                //wait for a call from a client
                client = server.accept();

                //start a new HelloServer for this client
                new HelloServer(client);
            }
        } catch (Exception e) {
            //No error handling - it's only an example ;- )
        }

        Thread serverThread;
        Socket sock;

        public HelloServer(Socket sock) {
            this.sock = sock;
            serverThread = new Thread(this);
            serverThread.start();
        }

        public void run() {
            try {
                //Say hello, wait a bit and then disconnect
                PrintWriter out =
                    new PrintWriter(sock.getOutputStream());

                out.println("Hello from Tini!");
                Thread.currentThread().sleep(1000);
                out.close();
                sock.close();
            } catch (Exception e) {
                //No error handling - it's only an example ;- )
            }
        }
    }
}
```


remember; you'll be using the special TINI VM so special core libraries are needed.

This should produce a file called HelloServer.class. Now we need to convert this class file into a .tini file,

```
java -classpath <TINIPATH>\tini.jar
TINIConvertor -f HelloServer.class -o
HelloServer.tini -d
<TINIPATH>\firmware\tini.db
```

again replacing the <TINIPATH> tag with the location of your TINI installation.

This command should produce a HelloServer.tini file. You will now need to FTP this class onto your TINI board. Windows and Linux both include command-line FTP clients, or you might like to use something like CuteFTP.

Nearly there! Telnet to your TINI board:

```
C:\>telnet 192.168.0.100
```

(use the standard username root and the password tini)

After logging in, start the HelloServer with the command

```
java HelloServer.tini &
```

Now open another telnet window and connect to your TINI on port 1234:

```
C:\>telnet 192.168.0.100 1234
```

TINI should say Hello and then, after a second or so, disconnect you.

Until next time

That's all for now. Next time, I'll be looking at some of the more useful features of TINI, including the various web-enabling technologies available. I will be showing how to hook up the one-wire iButton interface to a web-server, an RS232 terminal and an i/o port to create a simple web-enabled security system.

Resources

<http://www.ibutton.com/TINI>

<http://www3.sympatico.ca/guillaume.fournier/>

<http://java.sun.com/>

<http://www.apms.com.au/tini/>

TINI homepage - hardware, firmware, mail lists etc.

Excellent 'Getting Started' resource

Source of all things Java inc. JDK, javax.comm. required for use with TINI

Another good TINI resource site

SPECIAL OFFERS

TEKTRONIX 2445A

4 Channel 150MHz delay sweep, cursors/ readout
£750

TEKTRONIX 475

Dual Trace 200MHz delay sweep
£400

MARCONI 2610 TRUE RMS VOLTMEETER

Digital LCD + analogue meter 5Hz - 25MHz + DC autoringing
£195

AVO 8 Mk 6 MULTIMETER

In Ever Ready case with leads and batteries
£80
Other AVOs from **£50**

RACAL/AM 9343M LCR Databridge

Digital. £200

Auto measurements of R, C, L, Q, D. £125

HUNTRON TRACKER Model 1000 £380

FLUKE 8050A DMM 4 1/2 digit 2A True RMS. £75

FLUKE 8010A DMM 3 1/2 digit 10A. £50

FLUKE 8012A DMM 3 1/2 digit 2A. £40

RACAL TRUE RMS VOLTMEETERS

9300 5Hz-20MHz usable to 60MHz. 10 volts 316V **£95**

9300B version **£150**

GOODWILL GVT427 DUAL CHANNEL AC MILLIVOLTMETER

10uV 300V in 12 ranges

Frequency 10Hz-1MHz

Used £100 Unused £125

SINAD MEASUREMENTS

for only
MARCONI 893C AF POWER METER
300u watts-30 watts; 20Hz-35KHz;
2.5 Ohm - 20K Ohms **£60**

Marconi 893C unused, boxed **£100**

Marconi 893B - to 10 watts max. NO SINAD **£30**

GOULD J3B SINE/SQ OSCILLATOR

10Hz-1200KHz. Low distortion. Balanced metered output. Attenuator **£75-£125**

SOLARTRON 7150

DMM 6 1/2 digit. TRUE RMS. High accuracy IEEE **£95-£150**

HIGH QUALITY RACAL COUNTERS

9904 Universal timer counter 50MHz **£50**

9916 Counter 10Hz-520MHz **£75**

9918 Counter 10Hz-560MHz. 9 digit. **£50**

FARNELL AMM255

Automatic Mod Meter AM/FM 1.5MHz-2GHz 3.5 digit LCD Display. Unused **£400**

Also available: FARNELL AMM2000 Automatic Mod Meter 10Hz-2.4GHz. Unused **£950**

MARCONI 2305 Mod Meter 500KHz-2GHz from **£750**



MARCONI 2022E
Syn AM/FM Sig Gen 10KHz - 1.01GHz
Up to +10dBm output, phase mod. LCD display, keyboard entry etc. Small, lightweight
£525-£600-£750

RADIO COMMUNICATIONS TEST SETS

MARCONI 2955 £2000

MARCONI 2955A £2500

MARCONI 2019 Synth AM/FM Sig Gen 80KHz-104MHz £475

H.P. 8657A Synth Sig Gen 100KHz-104MHz £2500

H.P. 8656B Synth Sig Gen 100KHz-990MHz £1350

H.P. 8656A Synth Sig Gen 100KHz-990MHz £395

GIGATRONIC 7100 Synth Sig Gen 10MHz-20GHz £5000

MARCONI 2017 AM/FM Phase Locked Sig Gen 10KHz-1024MHz £1200

Good Signal Purity

H.P. 8640A AM/FM Sig Gen 500KHz-1024MHz £400

H.P. 8640A AM/FM Sig Gen 500KHz-512MHz £250

PHILIPS PM5328 Sig Gen 100KHz-180MHz with 200MHz Freq Counter IEEE £350

RACAL 9081 Synth AM/FM Sig Gen 5-520MHz £350

H.P. 3325A Synth Function Gen 21MHz £600

MARCONI 6500 Amplitude Analyser £1500

H.P. 4275A LCR Meter 10KHz-10MHz £2750

H.P. 8903E Distortion Analyser £750

WAYNE KERR Inductance Analyser 3245 £2000

H.P. 8112A Pulse Generator 50MHz £1250

DATRON AutoCal Multimeter 5 1/2 digit, 1065/1061A/1071 from £300-£500

MARCONI 2440 Frequency Counter 20GHz £1000

H.P. 5350B Frequency Counter 20GHz £2000

H.P. 5342A 10Hz-18GHz Freq Counter. £800

FARNELL AP10030 Power Supply. £1000

FARNELL AP7030 Power Supply. £800

PHILIPS PM5418TX Colour TV Pattern Generator £1750

PHILIPS PM5418TX1 Colour TV Pattern Generator £2000

B&K Accelerometer type 4366 £300

H.P. 11692D Dual Directional Coupler 2MHz-18GHz £1600

H.P. 11691D Dual Directional Coupler 2MHz-18GHz £1250

TEKTRONIX P6109B Probe 100MHz Readout. Unused £30

TEKTRONIX P6106A Probe 250MHz Readout. Unused £35

SPECTRUM ANALYSERS

H.P. 8562A 1KHz-22GHz £9000

ADVANTEST R3261A 9KHz-2.6GHz Synthesised £4000

EATON/AILTECH 757 0.001-22GHz £2500

TEKTRONIX 492 50KHz-18GHz £3500

H.P. 8558B with Main frame 100KHz-1500MHz £1250

H.P. 853A (Dig Frame) with 8559A 100KHz-21GHz £2750

MARCONI 2382 100Hz-400MHz High Resolution £2000

B & K 2033R Signal Analyser £1500

ADVANTEST TR4131 10KHz-3.5GHz £2750

MARCONI 2370 30Hz-110MHz from £500

HP141 Systems 8553 1KHz-110MHz from £500

8554 500KHz-1250MHz from £750; 8555 10MHz-18GHz from £1000

UNUSED OSCILLOSCOPES

TEKTRONIX TDS350 Dual Trace 200MHz 1GS/Sec £1500

TEKTRONIX TAS485 4 Channel 200MHz etc. £1100

H.P. 54600B Dual Trace 100MHz 20MS/s £1000

OSCILLOSCOPES

PHILIPS PM3092 2-2 Ch 200MHz Delay TB etc. £350

PHILIPS PM3082 2-2 Ch 100MHz Delay etc. £300

TEKTRONIX TAS465 Dual Trace 100MHz Delay etc. £800

TEKTRONIX 2465 4 Ch 300MHz Delay Sweep Cursors £1250

TEKTRONIX 2430 Dual Trace 150MHz 100MS/s Cursors etc. £800

TEKTRONIX 2232 Dual Trace 100MHz 100MS/s Cursors etc. £800

TEKTRONIX 2212 Dual Trace 60MHz 20MS/s Cursors etc. £550

TEKTRONIX 2210 Dual Trace 500MHz 20MS/s £450

H.P. 54200A Digitising 50MHz 200MS/s £600

PHILIPS PM3217 - Very good oscilloscope including 2 probes, Pouch and Front Cover

Dual Trace 50MHz Delay **£250-£300**

THIS IS THE BEST CHEAP SCOPE YOU WILL EVER BUY!!!

GOULD 051100 Dual trace, 30MHz delay, very bright. Supplied with manual and 2 probes. **£200**

STEWART of READING

110 WYKEHAM ROAD, READING, BERKS RG6 1PL

Telephone: (0118) 9268041 Fax: (0118) 9351696

Callers welcome 9am-5.30pm Monday to Friday (other times by arrangement)



CIRCLE NO.112 ON REPLY CARD

USED EQUIPMENT - GUARANTEED. Manuals supplied
This is VERY SMALL SAMPLE OF STOCK. SAE or telephone for lists. Please check availability before ordering. CARRIAGE all units £16. VAT to be added to total of goods and carriage.